

Virtualisoidut vs. fyysiset palvelimet sovellusaluksina

Tommi Laulainen

Opinnäytetyö
Helmikuu 2014

Tietotekniikan koulutusohjelma
Tekniikan ja liikenteen ala



JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES



Tekijä(t) Laulainen, Tommi	Julkaisun laji Opinnäytetyö	Päivämäärä 11.02.2014
	Sivumäärä 45 + 4	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty (X)
Työn nimi Virtualisoidut vs. fyysiset palvelimet sovellusalustoina		
Koulutusohjelma Tietotekniikan koulutusohjelma		
Työn ohjaaja(t) Rantonen, Mika Häkkinen, Antti		
Toimeksiantaja(t) Organisaatio		
<p>Tiivistelmä</p> <p>Opinnäytetyö tehtiin organisaation IT-osastolle. Työn tarkoituksena oli vertailla virtuaalisten ja fyysisten palvelimien käyttöä sovellusalustoina ja luoda dokumentti heidän käyttöönsä. Työssä tutkittiin virtuaalisten ja fyysisten palvelimien eroja teoriassa ja suorituskkyeroja käytännössä. Siinä käsiteltiin myös virtuaalialustojen pullonkauloja, kustannuseroja ja organisaation nykyistä ympäristöä.</p> <p>Pullonkaulojen osalta virtuaaliympäristössä suurimmaksi osottautui I/O-väylät, mikä johtui osaltaan käytetyistä x86-arkkitehtuuriin pohjautuvista palvelimista. x86-arkkitehtuuriin pohjautuvat virtuaalipalvelimet ovat hyviä suurta laskentatehoa kaipaaville sovelluksille, mutta eivät kovin hyviä I/O-intensiivisessä tiedonkäsittelyssä.</p> <p>Käytännön suorituskkytösteissä mitattiin virtuaalisten ja fyysisten palvelimien eroja yksinkertaisissa Java-pohjaisissa sovellusalustatesteissä. Niissä mitattiin viiveitä kysyvän ja kysytyn palvelimen välillä. Kysyvänä palvelimena toimi virtuaalipalvelin ja kysyttävänä palvelimena fyysinen tai virtuaalinen palvelin. Virtuaaliset palvelimet sijaitsivat joko samalla tai eri alustapalvelimilla. Lopuksi mitattiin kuinka nopeasti virtuaalinen ja fyysinen palvelin vastasivat itselleen.</p> <p>Mittauksissa selvisi, että suorituskkyvyn kannalta on väliä sijaitsevatko kaksi toisiinsa useasti yhteydessä olevaa virtuaalipalvelinta samalla vai eri alustapalvelimella. Fyysinen palvelin myös voitti virtuaalipalvelimen melko selvästi puhtaassa suorituskkyvyssä.</p>		
Avainsanat (asiasanat)		
I/O, JBoss, virtualisointi, VMWare, WAS, x86		
Muut tiedot		



Author(s) Laulainen, Tommi	Type of publication Bachelor's Thesis	Date 11022014
	Pages 45 + 4	Language Finnish
		Permission for web publication (X)
Title Virtualized vs. physical servers as software platforms		
Degree Programme Information Technology		
Tutor(s) Rantonen, Mika Häkkinen, Antti		
Assigned by Organization		
<p>Abstract</p> <p>The bachelor's thesis was assigned by an IT department of an organization. The purpose of the thesis was to compare the use of physical and virtualized servers as software platforms and to create a document for the organization. Differences between the two are dealt with in theory and also in practice with performance tests. The thesis also addresses the bottlenecks and cost differences in virtual environments and the organization's current environment.</p> <p>The bottlenecks in the virtual environment proved to be mostly I/O buses partly caused by x86 architecture-based servers. x86 architecture-based virtual servers are efficient when the applications only require computing power; however, they are not very good at intensive I/O operations.</p> <p>The performance tests measured the differences between virtual and physical servers in simple Java-based application tests. They measured delays between a server doing an inquiry and the server responding to it. The server carrying out the inquiries was a virtual server and the responding server was either virtual or physical. The virtual servers were either on the same platform servers or on different ones. The last tests measured how fast virtual and physical servers responded to themselves.</p> <p>The tests showed that it is of significance whether the two virtual servers inquiring each other are on the same physical platform server or not. The physical server also outdid the virtual server in pure computing power.</p>		
Keywords I/O, JBoss, virtualization, VMWare, WAS, x86		
Miscellaneous		

Sisältö

Lyhenteet ja termit	5
1 Työn lähtökohdat	7
1.1 Toimeksianto ja tavoitteet	7
2 Palvelintyypit	8
2.1 Fyysinen.....	8
2.2 Virtuaalinen	9
2.2.1 Virtuaalisen palvelimen osat	9
2.2.2 Virtualisointiohjelmisto	10
2.2.3 Virtuaalikone ja käyttöjärjestelmä.....	10
2.2.4 VMware vSphere	10
3 Virtualisoinnin edut	12
4 Mahdolliset pullonkaulat	13
4.1 CPU	13
4.2 RAM.....	13
4.3 I/O	14
4.3.1 Yleistä	14
4.3.2 NAS.....	15
4.3.3 SAN.....	15
4.4 Verkkolaitteet	16
4.5 x86-arkkitehtuuri	16
5 Toiminta vikatilanteessa	19
5.1 Yleistä	19
5.2 Fyysisen palvelinraudan rikkoutuminen	19
5.3 Virtuaaliympäristön vikatilanteet	20
5.3.1 Yleistä	20
5.3.2 VMware vSphere High Availability	20
5.3.3 VMware vSphere Fault Tolerance	21

5.3.4	Snapshotit.....	21
6	Sovellukset	22
6.1	Yleistä	22
6.2	Java-alustat.....	22
6.3	Laitevaatimukset.....	24
7	Kustannukset.....	25
7.1	Yleistä	25
7.2	Rauta	25
7.3	Open Source vs. kaupallinen sovellus	25
8	Toimeksiantajan nykyinen ympäristö	29
8.1	Yleistä	29
8.2	VMware	29
8.3	Fyysiset palvelimet.....	30
8.4	Levyjärjestelmät.....	31
9	Suorituskykymittaukset.....	33
9.1	Mittausväline	33
9.2	Testattavat palvelimet	34
9.3	Testausmetodit	35
9.4	Testit.....	35
9.4.1	Kysyvä ja kysyttävä palvelin eri alustapalvelimilla	35
9.4.2	Kyselyt virtuaaliselta palvelimelta fyysiselle	36
9.4.3	Kyselyt saman alustapalvelimen sisällä VM:ltä toiselle	37
9.4.4	Kyselyt VM:n sisällä.....	39
9.4.5	Kyselyt fyysisen palvelimen sisällä.....	39
9.5	Tuloksien pohdinta.....	40

10 Yhteenveto	42
10.1 Työn tulokset ja niiden arviointi	42
10.2 Kehittämiskohteet	43
Lähteet	44
Liitteet	46
Liite 1. Virtuant Corpin tutkimuksen kustannuserot WAS ND 7:n ja JBoss EAP 5:n välillä	46
Liite 2. Prolificsin tutkimuksen kustannuserot WAS V8.5:n ja Jboss EAP v6:n välillä ..	48
Liite 3. Suorituskykymittauspöytäkirja	49

Kuviot

Kuvio 1. Fyysisen palvelimen malli.....	8
Kuvio 2. Virtuaalisen palvelimen osat	9
Kuvio 3. VMware vSpheren kerrokset.....	11
Kuvio 4. Tietokoneen komponentit	17
Kuvio 5. Välimuisti ja RAM.....	18
Kuvio 6. Java EE -palvelin ja containerit	23
Kuvio 7. VMware-klustereiden CPU- ja RAM-käyttö	30
Kuvio 8. EJB-testeri.....	33
Kuvio 9. Testattavien palvelimien sijainnit toisiinsa nähden	34
Kuvio 10. Ensimmäisen testin topologia	35
Kuvio 11. Toisen testin topologia.....	37
Kuvio 12. Kolmannen testin topologia	38

Taulukot

Taulukko 1. WAS ja JBoss hankinta-, ylläpito- ja tukihinnat.....	27
Taulukko 2. VMware-palvelimien komponentit	29
Taulukko 3. Fyysisten palvelimien tiedot	31
Taulukko 4. Kyselyviiveet eri alustapalvelimien välillä	36
Taulukko 5. Kyselyviiveet virtuaaliselta palvelimelta fyysiselle	37
Taulukko 6. Kyselyviiveet saman alustapalvelimen sisällä.....	38
Taulukko 7. Kyselyviiveet virtuaalikoneen sisällä	39
Taulukko 8. Kyselyviiveet fyysisen palvelimen sisällä	40

Lyhenteet ja termit

API	Application programming interface, ohjelmointirajapinta
CPU	Central Processing Unit, prosessori
EIS	Enterprise Information Systems tier, Java EE –palvelinmallin datataso
EJB	Enterprise JavaBeans, yksi Java EE –palvelinmallin ohjelmointirajapinoista
ESXi	VMWaren yrityskäyttöön tarkoitettu hypervisor-tuote
FATA	tai FC-ATA on adapteri, jolla saadaan halpa ATA- tai SATA-asema kiinni FC-väylään
FC	Fibre Channel, SAN-verkkojen käyttämä nopea verkkoteknologia
FT	vSphere Fault Tolerance, vSpheren vikasietoisuuden mahdollistava teknologia
HA	vSphere High Availability, vSpheren korkean saatavuuden teknologia
HTTP	Hypertext Transfer Protocol, selaimien ja palvelimien väliseen tiedonsiirtoon tarkoitettu protokolla
I/O	Input/Output, tiedon siirtämistä komponenttien välillä
IOPS	Input/Output Operations Per Second, tapa esittää tallennusmedioiden I/O-suorituskykyä
JSP	JavaServer Pages, teknologia, jolla voidaan luoda dynaamisia web-sivuja
JVM	Java Virtual Machine, virtuaalikoneena käyttäytyvä sovellus, joka suorittaa Java-koodia
LAN	Local Area Network, lähiverkko
NAS	Network Attached Storage, verkossa oleva tallennuslaite

NIC	Network Interface Card, verkkokortti
RAC	Real Application Clusters, Oracle Databasen ominaisuus, joka mahdollistaa klusteroinnin ja korkean saatavuuden
RAID	Redundant Array of Independent Disks, teknologia, jolla yhdistetään kaksi tai useampi fyysinen kiintolevy yhdeksi tai useammaksi loogiseksi asemaksi. Tällä saavutetaan vikasietoisuutta ja nopeutta.
RAM	Random-Access Memory, keskusmuisti
SAN	Storage Area Network, erillinen tallennustilaa tarjoava verkko
SAS	Serial Attached SCSI (Small Computer System Interface), palvelinympäristöissä käytetty kiintolevyliitäntä
TCO	Total Cost of Ownership, arvio, joka auttaa ostajia ja omistajia ottamaan selville tuotteen tai järjestelmän suoria ja epäsuoria kustannuksia
VM	Virtual Machine, virtuaalikone
VMM	Virtual Machine Monitor, hypervisorin sisällä pyörivä monitori, joka vastaa resurssien jakamisesta virtuaalikoneille
WFC	Windows Failover Clustering, Windows Servereille tarkoitettu korkean saatavuuden takaava teknologia
WWW	World Wide Web

1 Työn lähtökohdat

1.1 Toimeksianto ja tavoitteet

Työn tarkoituksena oli tutkia ja vertailla virtuaalisten ja fyysisten palvelimien käyttöä sovellusalustoina sekä luoda näistä dokumentti toimeksiantajan käyttöön. Tavoitteena oli kehittää opiskelijan ja toimeksiantajaorganisaation henkilökunnan ymmärrystä virtuaalisista ja fyysisistä palvelimista sekä niiden hyvistä ja huonoista puolista sekä käyttökohteista. Dokumentin olisi tarkoitus olla hyödyllinen tulevassa uuden virtuaaliympäristön ja sovellusalustapalvelimien suunnittelussa ja hankinnassa.

Työssä käytiin läpi kustannuseroja niin rauta- kuin sovelluspuolella, mahdollisia pulonkauloja virtuaalisissa palvelimissa ja x86-arkkitehtuurissa yleensä sekä vikatilanteiden ja niihin varautumisen eroista. Näiden lisäksi käytiin Java-pohjaisia sovellusalustoja läpi ja kartoitettiin nykyistä ympäristöä sovellusalustojen näkökulmasta.

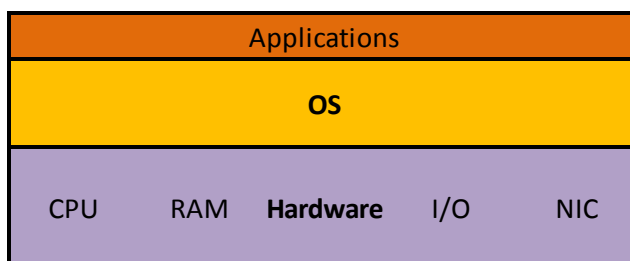
Työn käytännön osuus suoritettiin tekemällä käytännön suorituskykytestejä kahdella virtuaalipalvelimella ja yhdellä fyysisellä palvelimella. Testit toteutettiin organisaation työntekijän tekemällä Java-sovelluksella, jolla mitattiin kahden palvelimen välisessä kyselyssä kuluvaa aikaa. Testien tarkoituksena oli tuoda esille suorituskykyeroja virtualisoitujen ja ei-virtualisoitujen palvelimien välillä. Samalla tutkittiin, onko virtuaalipalvelinpuolella väliä sijaitsevatko kysyvät ja vastaavat palvelimet samalla vai eri alustapalvelimella.

2 Palvelintyypit

2.1 Fyysinen

Fyysisellä palvelimella tarkoitetaan tässä työssä sitä, että yhdelle fyysiselle palvelimelle on asennettu yksi palveluita tarjoava käyttöjärjestelmä, kuten Windows tai Linux. Välissä ei siis ole virtualisointiohjelmia, josta enemmän luvussa 2.2. Yhdellä fyysisellä palvelimella voi olla kuitenkin useampia palveluita, joita se tarjoaa. Käytettävistä resursseista kilpailevat siis pyöritettävät palvelut eli sovellukset, sekä itse käyttöjärjestelmä, joka myöskin jakaa resurssit niitä tarvitseville osapuolille.

Kuviossa 1 on esitetty fyysisen palvelimen malli ja kuinka käyttöjärjestelmä sijoittuu sovelluksiin ja rautaan nähden. Mallista selviää, että tämä on todella yksinkertainen ja helppo tapa toteuttaa palvelin. Käyttöjärjestelmällä on suora pääsy palvelimen resursseihin ilman välikättä eikä tästä synny overheadia, joten tällä tavalla raudasta saadaan irti paras ”maksimiteho”. Toisaalta taas resurssit todennäköisesti ovat joutilaana suurimman osan ajasta, jota hyvin toteutetulla virtualisoinnilla ei tapahdu.



Kuvio 1. Fyysisen palvelimen malli

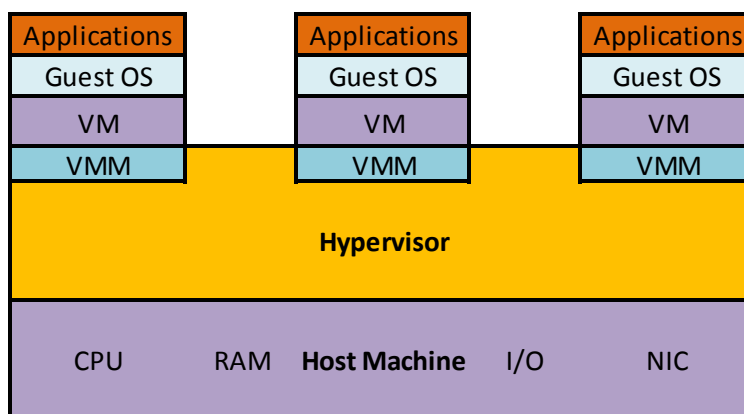
2.2 Virtuaalinen

2.2.1 Virtuaalisen palvelimen osat

Virtuaalinen palvelin koostuu neljästä perusosasta:

1. Host machine (fyysinen palvelin)
2. Hypervisor (virtualisointiohjelmisto)
3. Virtual Machine (VM, virtuaalikone)
4. Guest OS (Käyttöjärjestelmä).

Kuviossa 2 on hahmoteltu virtuaalisen palvelimen osat ja miten ne sijoittautuvat toisiinsa nähden. Host Machine on rauta, johon on asennettu virtualisointiohjelmisto (esim. VMWare ESXi). Tällä luodaan virtuaalikoneita, jotka näkyvät päälle asennettavalle Guest OS:lle fyysisinä koneina. Guest OS:n päällä pyörivät palvelut, joita halutaan tarjota kuten www- ja sähköpostipalvelimet. VMM (Virtual Machine Monitor) hoitaa resurssien jakamisen virtuaalikoneille, ja näitä resursseja ovat CPU (Central Processing Unit, prosessorit), NIC (Network Interface Controller, verkkokortit), I/O (Input/Output, kovalevyt, lisäkortit) sekä RAM (Random-Access Memory, muistit). (Herold, Laverick & Oglesby 2008, 17-21.)



Kuvio 2. Virtuaalisen palvelimen osat

2.2.2 Virtualisointiohjelmisto

Virtualisointiohjelmistoja on erilaisia: sellaisia, jotka asennetaan olemassa olevaan käyttöjärjestelmään kuin mikä tahansa muu sovellus, ja sellaisia, jotka asennetaan palvelimeen kuin käyttöjärjestelmä. Tässä työssä keskitytään vain jälkimmäiseen ja sitä kutsutaan nimellä *Hypervisor*. Tämä on tehokkain tapa virtualisoida palvelimia, koska hypervisorilla on suora pääsy resursseihin ilman välissä olevaa käyttöjärjestelmää. Näissäkin on muutamia eri toteutustapoja, mutta tässä työssä ei paneuduta niihin. (Understanding full virtualization, paravirtualization, and hardware assist, 2007.)

Hypervisorissa pyörivät monitorit, joita kutsutaan nimellä *Virtual Machine Monitor* (VMM), ovat vastuussa virtuaalikoneiden pyörittämisestä ja raudan virtualisoinnista niille. Jokaisen VMM:n saman hypervisorin sisällä täytyy käyttää samoja fyysisiä resursseja, jotta virtualisointi onnistuisi. (Understanding full virtualization, paravirtualization, and hardware assist, 2007.)

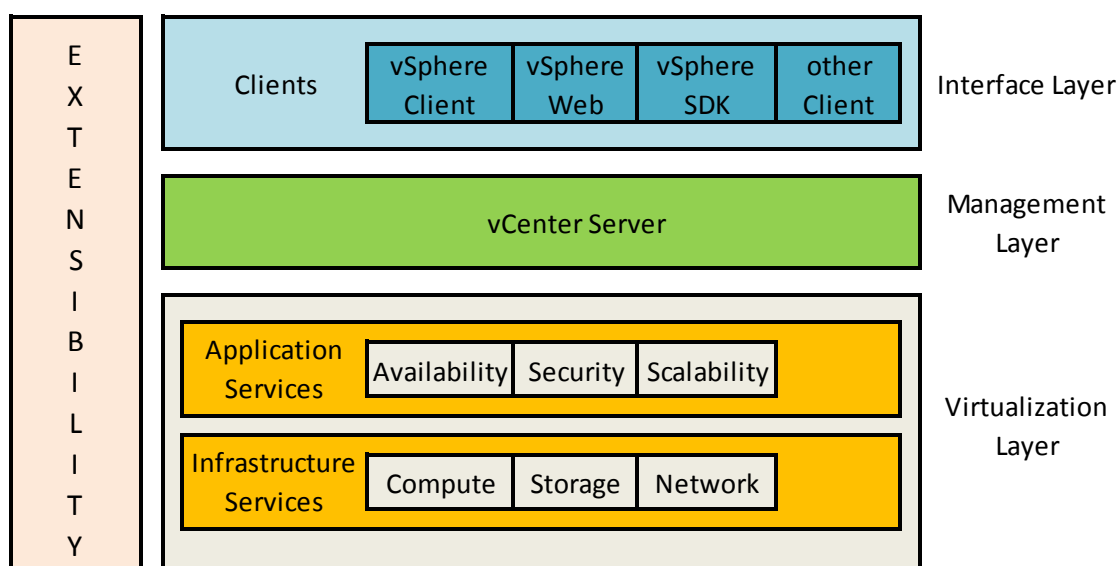
2.2.3 Virtuaalikone ja käyttöjärjestelmä

Virtualisoitavalle käyttöjärjestelmälle VM näkyy samanlaisena kuin normaali fyysinen rautapalvelinkin. Erona on kuitenkin se, että pohjimmiltaan Guest OS:lle näkyvät resurssit ovat vain looginen osa varsinaista rautaa. Virtuaalikäyttöjärjestelmät eivät kuitenkaan ole tietoisia toisista samalla raudalla pyörivistä palvelimista. Käyttöjärjestelmänä voi toimia mikä tahansa x86-pohjainen käyttöjärjestelmä aina MS-DOS:sta Windowseihin ja uusimpiin Linux-jakeluihin. (Herold ym. 2008, 23-24.)

2.2.4 VMware vSphere

VMware vSphere on alusta, jolla voidaan hallinnoida suuria infrastruktuureja saumattomina ja dynaamisina käyttöympäristöinä, kuten palvelimet, tallennusmedia ja verkko. Se koostuu kolmesta eri kerroksesta, joita ovat virtualisointi (virtualization), hallinta (management) ja käyttöliittymä (interface). Kuviossa 3 käy ilmi, mitä kukin kerros pitää sisällään ja kuinka laajennettavuus on jokaisella kerroksella läsnä. Virtualisointikerroksella sijaitsevat sovellus- (Application) ja infrastruktuuripalvelut, joista

ensimmäinen on vastuussa tarjottavien palveluiden saatavuudesta, jatkuvuudesta ja turvallisuudesta ja jälkimmäinen fyysisten resurssien jakamisesta palvelimien kesken. Hallintakerroksella on vCenter Server, joka on keskitetty hallintapiste koko virtualisointijärjestelmän monitorointiin, konfigurointiin ja hallinnoimiseen. Käyttöliittymäkerroksella ovat asiakasohjelmat (clientit), joilla voidaan mm. ottaa yhteys vCenter Serveriin ja sitä kautta hallinnoida koko järjestelmää. (VMware vSphere Basics n.d., 10-11.)



Kuvio 3. VMware vSpheren kerrokset

3 Virtualisoinnin edut

Yritykset haluavat tietojärjestelmiltä pienempiä kustannuksia ja häiriöaikoja sekä parempaa suorituskykyä ja skaalautuvuutta, jotta kustannukset pysyisivät mahdollisimman alhaisina. Virtualisointi on yksi tapa ratkaista tai ainakin edesauttaa kyseisten asioiden ratkaisemista. Campbellin ja Jeronimon (2006, 13-15.) mukaan virtualisoinnilla voidaan saavuttaa seuraavanlaisia etuja:

- Resurssien tehokkaampi käyttäminen
- Hallittavuus paranee
- Kokonaisvaltainen hallitseminen on yksinkertaisempaa
- Jokainen virtuaalikone on eristetty toisistaan sekä alla olevasta hypervisorista sekä jokaiselle virtuaalikoneelle voidaan määrittää erilaiset tietoturvaominaisuudet ja –asetukset. Näin tietoturvan taso kohenee.
- Vanhojen käyttöjärjestelmien, joille on hankala löytää yhteensopivia rautapalvelimia, migratoiminen virtualisoiduille palvelimille.
- Useiden, eri valmistajien käyttöjärjestelmien pyörittäminen samalla raudalla samanaikaisesti.
- Virtuaalikoneet ovat tallennettuina tiedostoihin, joten ne on helppo ja nopea varmistaa tai kopioida palvelimelta toiselle.
- Virtualisoinnilla voidaan ottaa helposti käyttöön esikonfiguroituja järjestelmiä.
- Virtuaalikoneet mahdollistavat kattavan ja helpon debuggauksen ja monitoroinnin ilman vaikutusta suorituskykyyn tai tarvetta erilliselle järjestelmälle.
- VM:ien eristyisyys toisistaan mahdollistaa vikojen etsimisen helposti ja turvallisesti. Tilojen tallennus, muokkaus ja palautus auttavat vian selvityksessä todella paljon.

4 Mahdolliset pullonkaulat

Virtualisoinnilla voidaan saada isotkin hyödyt, mutta se saattaa myös vaikuttaa palvelimien suorituskyykyyn heikentävästi. Osa heikentävistä tekijöistä on rautatasolla, ja tässä osiossa käydään ne läpi. CPU, fyysinen RAM, I/O sekä verkkokomponentit (itse palvelimessa sekä verkkoinfrastruktuurin laitteet) määräävät rautatasolla palvelimen suorituskyykyä ja ovat näin jokainen mahdollisia pullonkauloja virtuaalikoneille.

4.1 CPU

Ennen multi-core-prosessoreita CPU oli yksi suurimmista pullonkauloista virtualisoiduissa palvelinympäristöissä. Nykyään prosessorit ovat kehittyneet niin paljon, että ne ovat enää harvoin pullonkaulana, ja silloinkin ne on helppo korvata vain vaihtamalla tehokkaampaan tai enemmän ytimiä sisältävään malliin. Prosessorivalmistajat ovat myös lisänneet tuotteisiinsa virtualisointia helpottavia teknologioita, kuten AMD-V tai Intel VT-x, jotka siirtävät osan ohjelmiston työstä suoraan rautatasolle, jolloin tehokkuus kasvaa. Useamman ytimen prosessorit myös kuumenevat suhteutettuna huomattavasti vähemmän kuin yhden ytimen prosessorit ja näin vaativat vähemmän jäähdytystehoa. Tämä johtuu siitä, että kellotaajuksia ei ole tarvinnut nostaa vaan suorituskyykyä ollaan saatu lisäämällä ytimiä samalle mikropiirille. Suurempi kellotaajuus tarkoittaa suurempaa sähkövirtaa prosessorille ja näin korkeampaa lämpötilaa. (Virtualizing Server Workloads 2008.)

4.2 RAM

Keskusmuisti on I/O-järjestelmien kanssa isoin pullonkaula, ja muistin lisääminen verrattuna multi-core-prosessorin kustannuksiin on kalliimpaa. Pahimmillaan muistin lisääminen tarkoittaa uuden emolevyn hankkimista tai jopa kokonaan uuden palvelimen ostamista. Guest OS:n lisäksi keskusmuistia tarvitsee alla oleva hypervisor, eikä pelkästään itselleen vaan myös jokaista virtuaalikonetta kohden. Esimerkiksi, jos VM:lle annetaan 128 MB keskusmuistia, kokonaismuistintarve voi olla n. 200 MB,

johon sisältyy myös hypervisorin tarvitsema overhead. (Virtualizing Server Workloads 2008.)

4.3 I/O

4.3.1 Yleistä

Tietokonejärjestelmät tuottavat jatkuvasti suuria määriä dataa, ja kun siihen lisätään vielä virtualisointijärjestelmän tuottamat datat mukaan, päästään siihen lopputulokseen, että I/O on yksi suurimmista mahdollisista pullonkauloista. I/O-laitteisiin kuuluu levylaitteiden ja –järjestelmien lisäksi muitakin väyliä, kuten USB-väylät, mutta ne eivät ole olennainen osa I/O:ta tässä yhteydessä. Tallennusmediat virtuaalikoneissa voivat olla Host Machinen sisäisiä kovalevyjä, jolloin suorituskykyyn vaikuttavat itse kovalevyjen nopeudet ja niiden välimuistit, I/O-väylät ja -piirisarjat sekä mahdolliset RAID-konfiguraatiot. Toinen vaihtoehto on *Storage Area Network* (SAN) tai *Network Attached Storage* (NAS) –järjestelmä. (Virtualizing Server Workloads 2008.)

Toimeksiantajaorganisaation kokoisessa organisaatiossa käytännön tilanteissa pullonkaulana toimivat nimenomaan SAN-järjestelmät. Paikallisille kiintolevyille on asennettuna vain ja ainoastaan hypervisor-ohjelmisto eikä sinne tallenneta mitään muita tietoja. NAS-järjestelmät eivät taas tarjoa riittävää skaalautuvuutta, saatavuutta eikä suorituskykyä verrattuna SAN:lla toteutettuihin järjestelmiin.

Iso I/O-kuorma näkyy käyttäjille ja sovelluksille kasvavina viiveinä. Isoissa ympäristöissä suuret kuormat johtuvat mm. lokittamisesta ja tietokantakäsittelyistä. I/O-operaatiot kulkevat prosessorista eri väylien ja välimuistien (cache) kautta jopa levyjärjestelmien kovalevyille asti. Välissä voi olla vielä erilaisia tietoverkkoja, levyjärjestelmien levyohjaimia ja välimuisteja yms., joten pullonkaula voi olla missä tahansa näistä väylistä ja laitteista.

4.3.2 NAS

NAS on laite, joka toimii tallennusmedian lähiverkon yli työasemille, palvelimille tai molemmille käyttäen TCP/IP-protokollaa. Se tarvitsee älyä, jotta se voi käyttää järjestellä ja siirtää dataa, joten NAS on tavallaan yhdistelmä palvelinta, käyttöjärjestelmää ja tallennustilaa. Käyttäjä pääsee käsiksi NAS:lle tallennettuihin tietoihin käsi- mistä tahansa lähiverkon laitteelta, jolla on oikeudet. NAS on yleensä myös suojattu jollakin RAID-konfiguraatiolla, jotta sinne tallennettu tieto on vikasietoisempaa. On edullisempaa toteuttaa tämä keskitetyllä tallennuslaitteella kuin jokaisella työasemalla erikseen. (Bayo, Fuchs, Korn, Rebolj, Seidel & Tretau 2003 16.)

4.3.3 SAN

SAN on nopea verkko, joka yhdistää palvelimet ja tiedonvarastointilaitteet toisiinsa. Se hyödyntää mm. kytkimiä mahdollistaakseen *any-to-any*-yhteydet, joita ovat seuraavat:

- Server to storage: Tämä on ns. perinteinen tapa kommunikoida tallennuslaitteiden kanssa. Etuna SAN:ssa kuitenkin on, että samaa laitetta voidaan käyttää useiden palvelimien kesken peräkkäin tai samaan aikaan.
- Server to server: SAN:a voidaan käyttää nopeaan kommunikointiin palvelimien välillä.
- Storage to storage: Tämä mahdollistaa datan liikuttamisen tai kopioimisen SAN:n sisällä ilman, että palvelin tietää asiasta. Esimerkiksi tietoa voidaan tallentaa kasettilaitteelle pidempää säilytystä varten tai vaikka peilata toiselle laiteelle toiselle puolelle verkkoa, ja data on koko ajan palvelimien saatavilla.

SAN koostuu sitä käyttävistä palvelimista, jotka on liitetty *Host Bus Adapteereilla* (HBA) esimerkiksi *Fibre Channel* –kuituverkkoon ja sitä kautta itse varastointilaitteisiin. Se ei ota kantaa laitteiden käyttämiin tiedostojärjestelmiin, vaan jokainen palvelin voi alustaa sille allokoitun tilan haluamalleen tiedostojärjestelmälle. Tämä mahdollistaa saman tallennustilan jakamisen usean eri käyttöjärjestelmän kesken. (Beck, Ibarra, Kuvaravel, Miklas & Tate 2012, 11-12.)

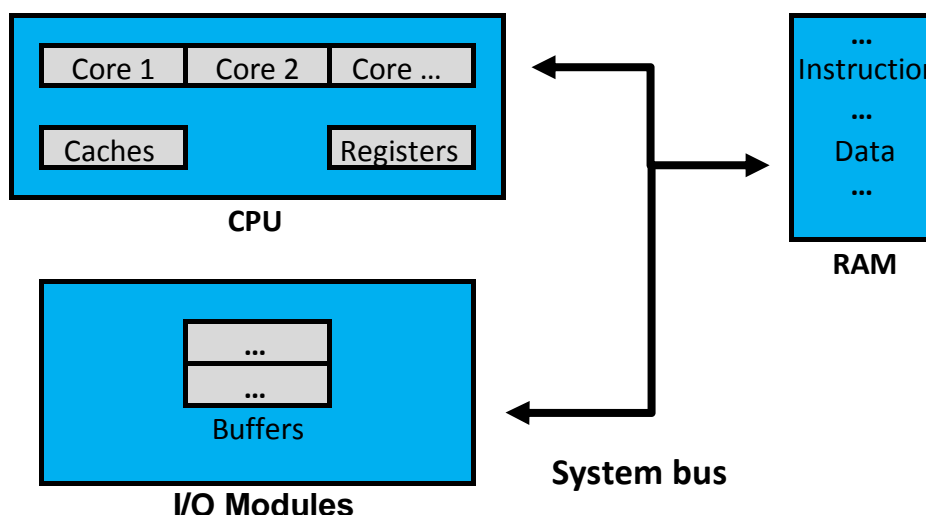
4.4 Verkkolaitteet

Vaikka verkkolaitteet eivät ole kovinkaan todennäköinen pullonkaula, on se silti mahdollinen, joten host-koneella täytyy olla tarvittava kapasiteetti virtuaalikoneiden verkkotarpeiden tyydyttämiseen. Verkon suorituskykyyn vaikuttavat palvelimien verkkokortit, lähiverkon laitteet kuten kytkimet ja reitittimet, lähiverkon kaapelointi sekä ulospäin tarjottaville palveluille myös internet-yhteyden ominaisuudet.

4.5 x86-arkkitehtuuri

Nykyprosessorien suuren laskentatehon ja prosessoriarkkitehtuurin välimuistiratkaisujen takia ideaalinen virtualisoitava kuorma on laskentakeskeistä. Kuitenkaan toimeksiantajaorganisaation tuottamat tehtävät palvelimille eivät ole sitä, vaan merkkijonojen käsittelyä sekä tiedostojen vastaanottoa, hakemista ja tallennusta. Tämän takia joitakin palvelimia on jouduttu siirtämään virtuaaliselta alustalta fyysisille palvelimille. Virtuaalipalvelimien ylisuuri I/O-kuorma hidasti muitakin palvelimia kohtuuttomasti, joten väliaikainen ratkaisu oli hankkia uusia palvelimia ja siirtää I/O:ta kuormittavat palvelimet niille. Toinen ratkaisu olisi ollut koko virtuaaliympäristön uusiminen, mutta sen suunnittelemiseen, toteuttamiseen ja testaukseen ei ollut aikaa. Nyt kuitenkin ollaan suunnittelemassa uusien alustapalvelimien hankkimista virtualisointikäyttöön.

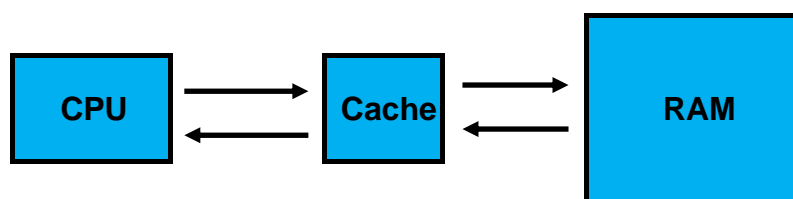
Tietokoneen voidaan ajatella koostuvan neljästä eri osasta: prosessorista (CPU), keskusmuistista (RAM), I/O-moduuleista ja järjestelmäväylästä (System bus). Kuviossa 4 on esitetty pääosin tietokoneen komponentit ja niiden suhteet toisiinsa. Prosessori koostuu mm. laskentayksiköistä (Core), välimuisteista (cache) ja rekistereistä (register). Prosessori on järjestelmäväylän kautta yhteydessä keskusmuistiin sekä I/O-moduuleihin, jotka sisältävät kaikki ulkoiset liitännät (prosessorin näkökannalta), joita ovat mm. kovalevyt, verkkokortit, hiiret ja näppäimistöt. (Stallings 2009, 8-9.)



Kuvio 4. Tietokoneen komponentit

Raa'an laskentatyön lisäksi prosessorin tehtäviin kuuluu tiedon vaihtaminen muistien kanssa. Koska prosessorien nopeudet ovat kasvaneet huomasti verrattuna keskusmuistien nopeuksiin, on jouduttu tekemään kompromisseja muistien suhteen. Tähän ongelmaan on ratkaisuna ollut jo pitkään hierarkinen muistirakenne, jossa lähimpänä laskentayksikköä ovat nopeimmat talletusyksiköt eli rekisterit. Seuraavana ovat prosessorin sisäiset välimuistit: L1-, L2- ja joissain uusimmissa prosessoreissa L3- välimuistit. Näiden koot ja sijainnit prosessorin sisällä vaihtelevat valmistajien ja mallien välillä. Välimuistit ovat kalliita, eikä niillä saavuteta isoja kapasiteetteja, joten ne eivät voi olla keskusmuistin korvaajia. (Stallings 2009, 26-33.)

Kuviossa 5 on esitetty välimuistin ja keskusmuistin konsepti: prosessori etsii ensin pienestä, mutta todella nopeasta välimuistista tietoa. Jos sieltä ei löydy, joudutaan turvautumaan prosessoriin verrattuna hitaaseen, mutta isoon keskusmuistiin. Välimuistin korvausalgoritmit päättävät, mikä tieto poistetaan välimuistista, jotta keskusmuistista haettu tieto saadaan tallennettua välimuistiin ja sitä kautta vietyä prosessorille suoritettavaksi. Prosessorin ja välimuistin välillä tieto liikkuu tavu kerrallaan, kun välimuistin ja keskusmuistin välillä data liikutetaan bloqueina, jotka koostuvat useista tavuista. (Stallings 2009, 26-33.)



Kuvio 5. Välimuisti ja RAM

5 Toiminta vikatilanteessa

5.1 Yleistä

Vikatilanteet ovat väistämätön paha tietotekniikan parissa, ja niihin onkin varauduttava aina. Täysin vikasietoista järjestelmää ei ole mahdollista luoda, mutta hyvin lähelle voidaan päästä. Mitä vikasietoisemmaksi pyritään, sitä kalliimmaksi se tulee, joten onkin järkevää löytää organisaation tarpeisiin ja budjettiin sopiva ratkaisu.

Viat voivat olla sovellus-, käyttöjärjestelmä- tai rautatasolla ja virtualisoiduissa ympäristöissä virtualisointitasolla. Jokaiselle tasolle on kehitetty erilaisia tekniikoita ja teknologioita, jotka tarjoavat korkeaa saatavuutta. Esimerkiksi sovellustasolla voisi olla *Oracle Real Application Clusters (RAC)*, käyttöjärjestelmätasolla *Windows Failover Clustering (WFC)*, virtualisointitasolla *vSphere High Availability (HA)* ja *vSphere Fault Tolerance (FT)* sekä fyysisellä tasolla redundanttisten komponenttien käyttäminen. Kuitenkaan yhden tason saatavuuden varmistaminen ei takaa saatavuutta toisella. Tämän takia olisi hyvä käyttää varmistuksia usealla eri tasolla ja tehdä suunnitelmat ja toteutukset organisaation tarpeiden mukaan. (Lowe 2011, 355-356.)

5.2 Fyysisen palvelinraudan rikkoutuminen

Jos itse palvelinrauta menee rikki tai siinä on vika, voi palvelimen kunnostamisessa mennä kauan. Tämä on tietenkin aina tapauskohtaista, mutta pääsääntöisesti palvelimen toimintakuntoon saattaminen voisi sisältää seuraavia muuttujia:

- Vianetsintä
- Uuden tai korvaavan osan tai palvelimen hankkiminen
- Uuden tai korvaavan osan tai palvelimen vaihtaminen

Jos vielä osio tai levy, jossa käyttöjärjestelmä sijaitsee, korruptoituu tai menee kokonaan rikki, joudutaan edellisien kohtien lisäksi käymään seuraavat läpi:

- Käyttöjärjestelmän asentaminen ns. tyhjältä pöydältä tai varmuuskopiosta sekä sen mahdollinen konfiguroiminen

- Sovellusten ja palveluiden asennus sekä konfiguroiminen

Palvelimen saamisessa toimintakuntoon voi kestää parista tunnista viikkoihin, riippuen suurimmaksi osaksi varaosien tai uuden palvelimen hankkimisesta. Kriittiset palvelut onkin järkevä vähintään kahdentaa tällaisten ikävien tapauksien varalta, jotta palvelu jatkuu, vaikka yksi palvelin ei olisi käyttökunnossa. Palvelimen käyttöjärjestelmälläkin on merkitystä, sillä Windowsin asennuksen pystyy palauttamaan helposti luodusta image-tiedostosta, kun taas Linuxin joutuu asentamaan alusta asti uudelleen.

5.3 Virtuaaliympäristön vikatilanteet

5.3.1 Yleistä

Virtuaaliympäristön palvelimet ovat yleisesti huomattavasti järeämpiä kuin yksittäiset fyysiset palvelimet, koska yhdellä raudalla pyörii useampi palvelin samanaikaisesti. Tämä luo sellaisen ongelman, että yhden fyysisen rauta-alustan rikkouduttua saatetaan monta palvelinta ja palvelua mennä alas yhtä aikaa. Tähän ongelmaan on ratkaisuja sekä käyttöjärjestelmätasolla että virtualisointitasolla. Ylimääräisillä resursseilla saadaan aikaiseksi vikasietoisempi ympäristö käytettäessä esimerkiksi vSpheren HA:ta tai FT:tä.

5.3.2 VMware vSphere High Availability

VMware vSpheren HA-ominaisuus mahdollistaa VM:n automaattisen uudelleenkäynnistämisen toiselle ESXi-koneelle sillä hetkellä, kun se ei ole enää saavutettavissa. Sen ensisijaisena tarkoituksena on hoitaa ESXi-koneen virheet, mutta sillä voidaan myös estää käyttöjärjestelmä- ja sovellustason virheitä. Kaikissa tapauksissa vSphere HA hoitaa ongelman VM:n uudelleenkäynnistämällä ja tästä syntyy pieni käyttökatko. Katkon pituus riippuu monista tekijöistä, joten sen pituutta ei voida etukäteen tarkalleen tietää, mutta se voi pienimmillään olla minuutteja. (Lowe 2011, 371-384.)

HA:ta käytettäessä tiedon korruptoituminen on mahdollista, joskin erittäin epätodennäköistä nykyisten käyttöjärjestelmien tiedostojärjestelmillä. Jos vika on tarpeek-

si suuri eikä resursseja ole vapaana tarpeeksi kaikkien virtuaalikoneiden pyörittämiseen, jää osa palvelimista käynnistymättä. Valinnaisena ominaisuutena on myös jo käynnissä olevien matalan prioriteetin palvelimien sammuttaminen suurempien prioriteetin palvelimien edestä pois. (Lowe 2011, 371-384.)

5.3.3 VMware vSphere Fault Tolerance

VMware vSpheren Fault Tolerancella (FT) saavutetaan palvelun jatkuva saatavuus. Se käyttää VMwaren vLockstep-teknologiaa, joka pitää ensisijaisen ja toissijaisen palvelimen lockstepissä. Tämä tarkoittaa sitä, että kun ensisijaiselle palvelimelle annetaan käskyjä, esimerkiksi hiiren liikuttamista ja klikkauksia, ne menevät täsmälleen samantyyppisinä toiselle palvelimelle, joka suorittaa käskyt kuin ominaan. Molempien toimiessa vain ensisijainen palvelin vastaa verkkokyselyihin, ja vikatilanteen sattuessa ottaa toissijainen palvelin tehtävän itselleen, eikä näin palvelukatkoa synny. Huonona puolelta tässä kuitenkin on se, että ensisijaisen palvelimen käyttöjärjestelmävirhetilanne kopioituu täsmälleen samantyyppisinä toiselle palvelimelle. Tällä siis saavutetaan vain vikasietoisuus yhtä tai useampaa ESXi-palvelimen vikaantumista vastaan. (Lowe 2011, 393-399.)

5.3.4 Snapshotit

VMware vSpheren VM-snapshotit mahdollistavat virtuaalikoneiden tilojen tallennuksen ja niiden palauttamisen. Kun virtuaalikoneesta otetaan snapshot, siitä tallentuu levyjärjestelmään tiedosto, joka kuvaa VM:n tilaa snapshotinottohetkellä. Tämän lisäksi virtuaalikoneesta voidaan tallentaa myös keskusmuistin sisältö erilliseen tiedostoon. Snapshotit eivät suojaa virtuaalikoneen virheiltä vaan lähinnä käyttöjärjestelmä- ja sovellustasolla olevilta virheiltä, eikä niitä voida käyttää varmuuskopiointikeinona. Niillä voidaan esimerkiksi ottaa nopea tilantallennus ennen päivityksen asentamista käyttöjärjestelmään, jolloin asennuksen mahdollisesti epäonnistuessa voidaan palauttaa tila ennen virhettä tai suojautua huonosti käyttäytyviä sovelluksia tai prosesseja vastaan. (Lowe 2001, 498-503.)

Toisaalta toimeksiantajan ympäristössä snapshotteja voidaan käyttää palvelimien varmuuskopiointikeinona, sillä tuotantoympäristössä tapahtuu niin vähän muutoksia.

6 Sovellukset

6.1 Yleistä

Toimeksiantajaorganisaation palvelimilla pyörii lukuisia sovelluksia, joista tässä työssä käydään läpi Java-alustat ja niihin liittyvät sovellukset.

6.2 Java-alustat

Javalla tarkoitetaan sekä ohjelmointikieltä että alustaa. Java-alusta on ympäristö, jossa ajetaan Javalla kirjoitettuja ohjelmia, joita on neljä:

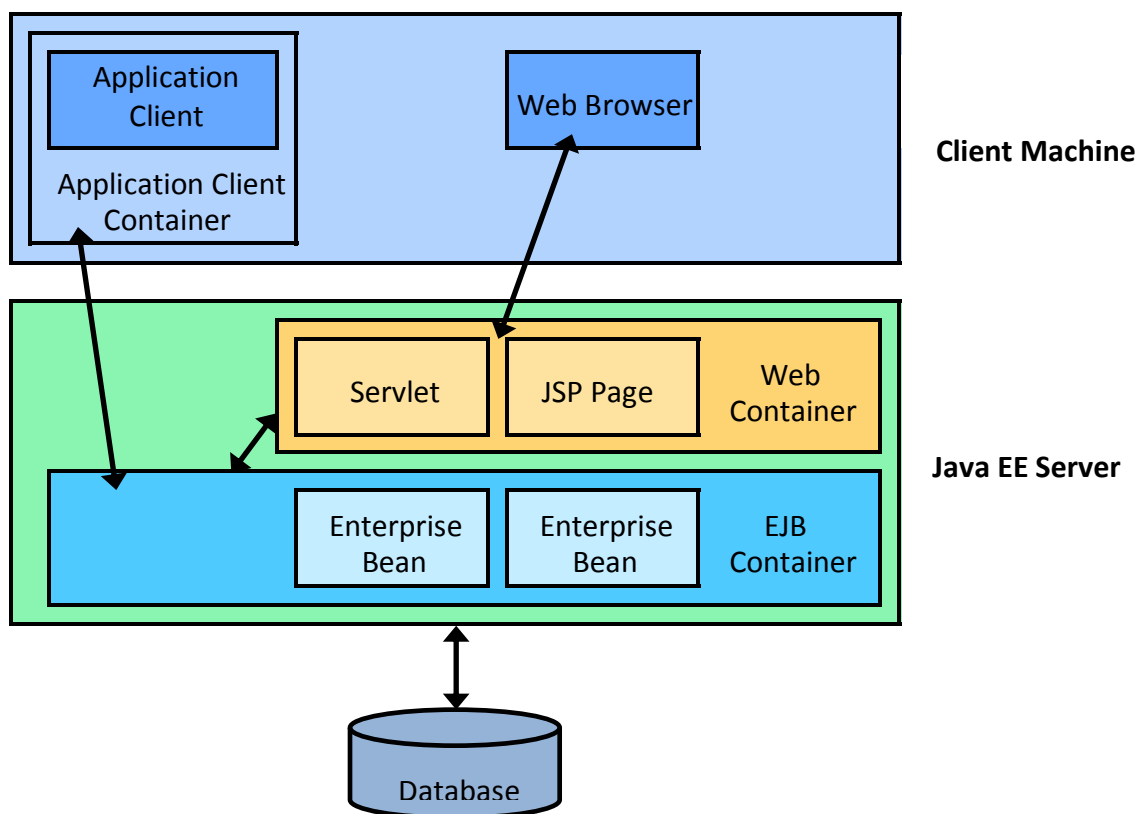
- Java Platform, Standard Edition (Java SE)
- Java Platform, Enterprise Edition (Java EE)
- Java Platform, Micro Edition (Java ME)
- JavaFX.

Kaikki alustat sisältävät *Java Virtual Machinen* (JVM) ja *Application Programming Interfacen* (API). Näistä ensimmäinen on ohjelma tietylle rauta- ja sovellusalustalle, joka pyörittää Javalla tehtyjä ohjelmia ja jälkimmäinen on kokoelma ohjelmistokomponentteja, joilla voidaan luoda uusia ohjelmistokomponentteja tai ohjelmia. Java SE:n API tarjoaa Java-ohjelmoinnin perustoiminnallisuudet ja määrittää ohjelmointikielen perustyyppit, -objektit sekä korkean tason luokat. Näiden lisäksi se sisältää mm. JVM:n sekä kehitys- ja käyttöönotto työkalut. Java EE sisältää edellä mainittujen lisäksi ajoympäristön suuremmille sovelluksille. (Evans 2013.)

Java EE mahdollistaa esimerkiksi monitasoisen sovellusmallin, joka yleensä sisältää asiakastason (client tier), keskitason (middle tier) ja datatason (*Enterprise Information Systems tier*, EIS). Asiakastasolla on asiakasohjelma, jolla tehdään pyyntöjä keskitasolle. Se voi olla erillinen ohjelma, toinen palvelin tai vaikka nettiselain, johon on asennettu Java. Keskitason tehtävänä on prosessoida käyttäjän tekemä pyyntö ja tallentaa tiedot pysyvästi datatasolle tietokantaan sekä mahdollisesti palauttaa käyttäjälle tietoa. (Evans 2013.)

Java EE –palvelimia kutsutaan monesti sovellusalustoiksi, koska ne tarjoavat sovellusdataa asiakasohjelmille, kuten web-palvelimet tarjoavat web-sivuja selaimille. Kuviossa 6 on esitetty yksinkertaisesti Java EE -palvelimen eri komponentit ja niiden suhteet toisiinsa. Asiakaskoneessa Java EE -palveluita käytetään joko erillisellä sovelluksella tai sitten normaalilla nettiselaimella, jossa on Java-tuki. Eri containerit tarjoavat niiden sisällä pyöriville komponenteille ajoympäristön. (Java EE 5 Tutorial 2007.)

Tavallinen nettiselain tekee pyyntöjä palvelimen Web Containerin sisällä oleville komponenteille, joita ovat esim. servletit ja JSP-sivut, ja erillinen asiakasohjelma voi olla suoraan yhteydessä EJB Containerin komponentteihin eli Enterprise Beansiin. Pyynnön jälkeen EE-palvelimen komponentit ovat yhteydessä toisiinsa ja mahdollisesti tietokantaan sekä luovat dynaamista sisältöä, joka palautetaan asiakassovellukselle tavallisena http-vastauksena. Kaikki logiikka on toteutettu siis palvelimessa, minkä takia asiakassovellukset voivat olla hyvinkin kevyitä ja niitä voidaan käyttää heikkotehoisilla laitteilla kuten mobiililaitteilla. (Java EE 5 Tutorial 2007.)



Kuvio 6. Java EE -palvelin ja containerit

6.3 Laitevaatimukset

Laitevaatimuksia sovellukset asettavat prosessoriarkkitehtuurin, käyttöjärjestelmän, muiden sovellusten ja palvelinraudan suhteen. Yleensä näistä annetaan vähintään minimivaatimukset ja joskus lisäksi myös suositeltavat laitekoonpanot. Rautavaatimuksia on hankala määrittää palvelimissa käytettäville skaalautuville sovelluksille, sillä järjestelmien laajuus voi vaihdella niin suuresti eri organisaatioiden verkoissa.

IBM määrittelee WAS ND 8.5.5:n laitevaatimukset käytettävän käyttöjärjestelmän ja WAS:n komponentin mukaan. Näiden lisäksi määritellään yhteensopivat Java-, web-palvelin- ja muut ohjelmistoversiot. Minimirautavaatimukset Linuxilla pyörittämiseen WAS ND:lle ovat 2 GB kovalevytilaa asennetulle imagelle ja väliaikaistiedoille tai 3 GB alustariippumattomalle asennukselle, 1 GB keskusmuistia ja 1,2 GHz Pentium 4 tai uudempi prosessori. (WAS system requirements 2013.) Red Hat ei anna rautavaatimuksia uusimmalle JBossille, joka on kirjoitushetkellä EAP 6.2, vaan ainoastaan yhteensopivat käyttöjärjestelmät, Java-versiot sekä muut sovellukset.

7 Kustannukset

7.1 Yleistä

Palvelimien kokonaiskustannukset koostuvat monista eri tekijöistä. Näitä ovat palvelinraudan hankinta, sähkö, jäähdytys, käyttöjärjestelmien ja sovelluksien lisenssit sekä erilaiset huolto- ja ylläpitosopimukset.

7.2 Rauta

Palvelinraudan hinta koostuu monista pikkutekijöistä, kuten emolevystä, prosessoreista, muisteista, kovalevyistä ja mahdollisista lisäkorteista. Näiden lisäksi laitevalmistajat tarjoavat erilaisia takuu- ja huoltopalveluita. Fyysisinä, yksittäisinä palvelimina käytettävät laitteet ovat huomattavasti halvempia kuin virtuaalialustoiksi tulevat palvelimet. Toisaalta taas virtuaalialustat ovat todella paljon järeämpiä ja osaavat virtualisoinnin takia käyttää resursseja paremmin hyödykseen, koska toimivat useana loogisena palvelimena. Tämän takia hinta palvelinta kohden voi olla huomattavasti edullisempaa kuin yksittäisiä fyysisiä palvelimia ostaessa.

7.3 Open Source vs. kaupallinen sovellus

Vapaan lähdekoodin, eli Open Sourcen, ohjelmistojen hinta koostuu lähinnä ylläpito- ja tukikustannuksista itse sovelluksen ollessa ilmainen, vapaasti muokattavissa ja käytettävissä. Kaupalliset sovellukset ovat taas suljettuja ja niiden hinta koostuu edellisten lisäksi myös lisenssien hankintakustannuksista, jotka voivat olla todella suuret. Yleensä lisenssejä on erilaisia, joilla on rajoitettu ohjelmiston toimintaa ja ominaisuuksista joudutaan maksamaan vielä erikseen. Tämän lisäksi lisenssin hintaa voi kasvattaa myös ympäristön koko, käyttäjien määrä tai laitteiston resurssit, mihin sovellusta ollaan ottamassa käyttöön.

Virtuant Corpin laatimassa tutkimuksessa (2011) vertailtiin IBM:n WebSphere Application Server Network Deployment 7:n ja Red Hatin JBoss Enterprise Application Platform 5:n välisiä kustannus-, ominaisuus- sekä suorituskykyeroja. Kyseiset sovel-

lukset ja niiden versiot ovat ominaisuuksiltaan toisiaan vastaavat sekä kiinnostavia toimeksiantajan kannalta. Kirjoitushetkellä uusimmat versiot sovelluksista ovat WAS 8.5.5 ja JBoss 6.1.0.

Virtuant Corpin tutkimuksessa kustannuserot WAS ND 7:n ja JBoss EAP 5:n välillä kallistuivat rajusti JBossin suuntaan kokonaiskustannuserojen ollessa ensimmäisenä vuotena lähes 2 miljoonaa dollaria, eli 67,57 %, JBossin hyväksi ja kolmen vuoden ajalta reilu 2,5 miljoonaa dollaria, eli 50,91 %. Liitteessä 1 on suomennettu taulukko sovellusten välisistä kustannuksista eri kategorioissa. Kannattaa kuitenkin huomioida, että tutkimuksessa on käytetty JBossin asiantuntijoille ja sovelluskehittäjille alemmaa vuosipalkkaa. Tätä on perusteltu indeed.com –sivuston antamista keskipalkoista kyseisille työtehtäville Yhdysvalloissa sekä Flemingin ja Perryn (2010) tekemästä tutkimuksesta sovelluskehityksen säästöistä käytettäessä JBossin EAP:ia. Toinen huomiotava asia on, että Red Hat ei itse tarjoa suomenkielistä tukea, vaan se tulee joltain ulkopuoliselta yritykseltä. IBM puolestaan tarjoaa suomenkielistä tukea.

Taulukossa 1 on esitetty Virtuant Corpin tutkimuksen hankinta-, ylläpito- ja tukihinnat WAS ND 7:lle ja JBoss EAP 5:lle. Hinnoittelussa käytettiin 9 kpl palvelimia varustettuna kahdella Intelin Xeon E-5640 –4-ydinprosessoreilla. IBM hinnoittelee prosessorit *Processor Value Unit*:n (PVU) mukaan, missä korkeampi PVU tarkoittaa karkeasti tehokkaampaa laskentatehoa (PVU n.d.). Lisenssit hinnoitellaan prosessorin ytimien määrän mukaan ja esimerkiksi kyseiselle Intelin prosessorille PVU per ydin on 70 ja yhden PVU:n hinta on 184 dollaria, täten WAS ND:n hinnaksi per fyysinen palvelin tuli 103 040 dollaria.

JBossin lisenssit taas eivät maksa mitään, joten ero kahden välillä on huima. Ohjelmistotueksi IBM:n tuotteelle oletettiin 17,5 % sekä 5,0 % lisenssin hankintahinnasta vuosimaksuksi sekä vuosittain uusittavaksi tilaukseksi. JBossilta vertailuun otettiin JBoss EAP premium –tukipaketti 51 250 dollarin vuosihintaan 72 prosessoriytimelle. Ohjelmiston valvonnassa ja hallinnassa IBM:ltä valittiin Tivoli (for WebSphere), jonka hankintahinnaksi oletettiin 696 960 dollaria ja vuosimaksuksi 34 848 dollaria. JBossilta vastaava ohjelmisto sisältyy EAP:n tilaukseen eikä siitä synny erillisiä kustannuksia.

Taulukko 1. WAS ja JBoss hankinta-, ylläpito- ja tukihinnat

Kategoria	Osio	WAS ND 1. vuosi	3 vuoden TCO	Jboss EAP 1. vuosi	3 vuoden TCO
Lisenssit					
	Palvelimien lkm.	9		9	
	Hinta / palvelin	\$103,040		\$0	
	YHT	\$927,360	\$927,360	\$0	\$0
Ohjelmistotuki		% lisenssistä			
	Ylläpito	17.5 %		\$0	
	Vuositilaus	5.0 %		\$51,250	
	YHT	\$208,656	\$625,968	\$51,250	\$153,750
Monitorointi ja hallinta (Tivoli & JON)					
	Hankinta	\$696,960		\$0	
	Vuositilaus	\$34,848		\$0	
	YHT	\$731,808	\$801,504	\$0	\$0
	LOPPUSUMMA	\$1,867,824	\$2,354,832	\$51,250	\$153,750

Tutkimuksessa saadut tulokset näyttävät hurjilta avoimen lähdekoodin ohjelmiston hyväksi. Toisaalta tutkimuksen on tilannut Red Hat, joten kyseinen tulos saattaa olla hieman kyseenalainen. Siinä käytiin pikaisesti myös tuen toimiminen molemmilla osapuolilla ja keuhuttiin JBossin yhteisöjä sekä internetistä saatavan tiedon määrää. IBM:n kanssa tukiasiat jouduttiin hoitamaan puhelimen kautta, koska internetin kautta ei ilmeisesti löytynyt auttavaa materiaalia. Oman kokemukseni perusteella IBM:n puhelintuki on kuitenkin toimivaa sekä saatavilla suomeksi.

IBM:n Prolificsilla teettämässä tutkimuksessa (WAS V8.5 vs. Jboss EAP v6 TCO Analysis 2012.) viiden vuoden *Total Cost of Ownership* (TCO) kallistui melko selvästi WAS:n hyväksi. Sen suomennettu lopputulos löytyy liitteestä 2. Tutkimuksessa käytettiin uudempia versioita molemmista ohjelmistoista kuin Virtuant Corpin vastavassa. Tutkimuksissa yhteneväisyyttä oli molempien ohjelmistojen hankintahinnat, jotka olivat ylivoimaisesti JBossin kannalla. Molemmissa tutkimuksissa suorituskykytestit kallistuivat tutkimuksen teettämän tahon ohjelmiston puolelle, mutta vain Prolificsin tutkimuksessa se otettiin huomioon rautahankintojen kohdalla. Siinä myös väitettiin JBossin olevan sopimaton sovelluksille, jotka vaativat synkronointia monen datalähteen välillä. Tämän takia vertailussa arvioitiin viiden vuoden riskiksi useita miljoonia dollareita, kun taas WAS:n kohdalla riskit arvioitiin olemattomiksi.

Tutkimuksien lopputuloksiin on vaikea ottaa sen enempää kantaa, koska molemmat ovat selvästi niiden teettämien osapuolien ohjelmistojen kannalle kallistuneita. Ohjelmistojen hankintahinnat ovat kuitenkin selvästi JBossin etu ja säästyneillä miljoonilla voisi vaikka palkata muutaman pelkästään JBossiin erikoistuneen asiantuntijan. Sovelluksien ominaisuuksien ja suorituskykyerojen puolueeton testaaminen ja vertailu itsessään olisi jo ainakin yhden opinnäytetyön laajuinen asia, joten sitä ei tässä työssä käsitellä.

8 Toimeksiantajan nykyinen ympäristö

8.1 Yleistä

Toimeksiantajaorganisaation nykyisessä ympäristössä suurin osa palvelimista on virtualisoitu. Muutamia palvelimia toimii edelleen suoraan fyysisellä raudalla, koska virtuaalikoneissa ei riittänyt teho niiden pyörittämiseen ja pullonkaulana oli nimenomaan I/O.

8.2 VMware

Virtuaaliympäristössä organisaatiolla on käytössä HP c7000 korttipalvelinkehikot sekä A- että B-konesaleissa. Niissä on kahdennetut virtalähteet ja *Virtual Connect* (VC) – tekniikan mahdollistavat HP:n Flex 10 -verkkomodulit. Palvelimina toimivat 10 kpl HP:n ProLiant BL460c G6 ja 6 kpl BL460c G7 –korttipalvelimet, joista puolet toisessa konesalissa ja puolet toisessa. Palvelimien kokoonpanot käyvät ilmi taulukossa 2. Jokaisesta palvelimesta löytyy kaksi Intelin neljäytimistä prosessoria, jotka tukevat Hyper Threading –ominaisuutta, joka mahdollistaa jokaisen ytimen käyttämistä kahdena loogisena prosessorina. Yhteensä loogisia prosessoreja on siis 16 kpl palvelinta kohden. Jokainen palvelin on myös yhteydessä SAN-verkkoon kahdella FC-liitännällä.

Taulukko 2. VMware-palvelimien komponentit

	BL460c G6	BL460c G7
CPU	2x Intel Xeon 4-core 2,66 Ghz	2x Intel Xeon 4-core 3,06 Ghz
RAM	192 GB	192 GB
HDD	2x 146 GB	2x 72 GB
NIC	2x 10 GbE	2x 10 GbE
FC	2x 8 Gb	2x 8 Gb

Molemmissa konesaleissa on käytössä 5 kpl G6- ja 3 kpl G7-palvelimia ja tuo kahdeksan palvelimen kokonaisuus on laitettu toimimaan yhtenä klusterina. Yhdessä klusterissa on 1536 GB keskusmuistia ja 128 loogista prosessoria. Jokaiseen ESXi-hostiin on

konfiguroitu vMotion, HA ja FT –ominaisuudet päälle, joten virtuaalipalvelimien migratointi onnistuu vaikka klustereiden välillä nopeasti ja toisen klusterin kaatuessa ottaa toinen klusteri kaatuneet virtuaalipalvelimet pyöritettäväkseen. Tämän takia molemmissa klustereissa pitäisi olla n. puolet CPU ja RAM –resursseista vapaana.

Kuviossa 7 nähdään VMware-klustereiden tämän hetkinen prosessori- ja muisti-kuorma arkipäivänä. Kuvankaappauksesta ilmenee selvästi prosessoreiden vähäinen kuormitus verrattuna muistin käyttöön. CPU-kuorma voitaisiin ainakin tuplata turvalisesti, mutta muistinkäyttöä ei ole varaa kovin paljoa nostaa, jotta noin puolet resursseista olisi vapaana.

Name	State	Status	% CPU	% Memory	Memory Size	CPU Count	NIC Count
	Connected	✓ Nor...	13	49	196597,70 MB	2	8
	Connected	✓ Nor...	24	50	196597,70 MB	2	8
	Connected	✓ Nor...	15	53	196597,70 MB	2	8
	Connected	⚠ War...	11	49	196597,70 MB	2	8
	Connected	⚠ War...	19	49	196597,70 MB	2	8
	Connected	⚠ War...	0	1	196597,70 MB	2	8
	Connected	⚠ War...	12	47	196597,70 MB	2	8
	Connected	✓ Nor...	15	48	196597,70 MB	2	8

Name	State	Status	% CPU	% Memory	Memory Size	CPU Count	NIC Count
	Connected	✓ Nor...	25	46	196597,70 MB	2	8
	Connected	✓ Nor...	26	47	196597,70 MB	2	8
	Connected	✓ Nor...	14	49	196597,70 MB	2	8
	Connected	✓ Nor...	9	37	196597,70 MB	2	8
	Connected	⚠ War...	15	48	196597,70 MB	2	8
	Connected	✓ Nor...	19	47	196597,70 MB	2	8
	Connected	✓ Nor...	42	42	196597,70 MB	2	8
	Connected	⚠ War...	13	47	196597,70 MB	2	8

Kuvio 7. VMware-klustereiden CPU- ja RAM-käyttö

8.3 Fyysiset palvelimet

Sovellusalojen kannalta olennaisten fyysisten palvelimien ominaisuudet ovat esitetty taulukossa 3. Palvelimia on kolme erilaista, joista 2,13 GHz:n versioita on kolme kappaletta, 2,4 GHz:n versioita kaksi kappaletta ja yksi 3,2 GHz:n versio. Tehokaimmassa palvelimessa on myös 8 GB enemmän muistia kuin muissa. Kaikissa on yhtä nopeat verkkoliitännät ja SAN-liitännät. Uusimmissa (G8) palvelimissa on myös isommat kovalevyt kuin vanhemmissa (G7), mutta sillä ei ole suorituskykyyn tai

muuhun mitään vaikutusta, koska paikalliselle koneelle on asennettuna vain käyttöjärjestelmä ja sovellukset, itse data sijaitsee SAN-verkossa.

Taulukko 3. Fyysisten palvelimien tiedot

	BL460c G7	BL460c G8
CPU	Intel Xeon 4-core 2,13 Ghz	Intel Xeon 4-core 2,4 Ghz
RAM	24 GB	24 GB
HDD	2x 72 GB	2x 500 GB
NIC	2x 10 GbE	2x 10 GbE
FC	8 Gb	8 Gb

	BL460c G7
CPU	Intel Xeon 4-core 3,2 Ghz
RAM	32 GB
HDD	2x 72 GB
NIC	2x 10 GbE
FC	8 Gb

8.4 Levyjärjestelmät

Toimeksiantajalla on kirjoitushetkellä käytössään neljä kappaletta HP:n valmistamia EVA 8400 –levylaittejärjestelmiä. Molemmissa konesaleissa on kaksi kappaletta kyseisiä laitteita ja ne ovat peilattuja toisiinsa konesalien välillä. Peilaus tapahtuu synkronisesti HP:n *Continuous Access (CA)* –tekniikalla, joka osaltaan vaikuttaa heikentävästi I/O-nopeuteen palvelimien kannalta, koska palvelimelle annetaan kuittaus kirjoituksesta vasta datan ollessa molemmissa saleissa kirjoitettuna.

Laitteissa on 4 Gb FC-kuituliitännät erilliseen kahdennettuun SAN-verkkoon, jonka reunakytkimet toimivat joko 4 Gb tai 8 Gb –nopeuksilla, riippuen käytössä olevasta laitteesta. SAN-verkon runkokytkimet ovat 8 Gb –liitännöillä. EVA 8400:n koostuu pääsääntöisesti 3,5 tuuman, 600 GB:n, 15000 kierrosta / min, FC-liitännäisistä kovalevyistä, joilla säilytetään eniten käytössä oleva data. Näiden lisäksi käytössä on myös muutamia 3,5 tuuman, 1 TB:n, 7200 kierrosta / min, FATA-liitännäisiä, hitaita varasto-levyjä. Levyjärjestelmät ovat konfiguroitu käyttämään RAID-6 –teknologiaa, joka mahdollistaa kahden yhtäaikaisen levyrikon levylaitetta kohden. Toimeksiantajan

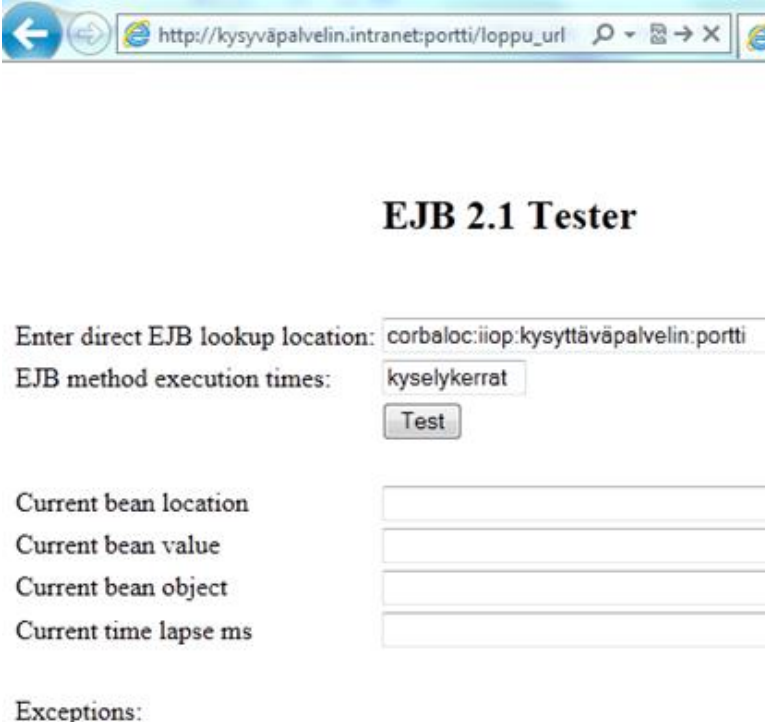
levylaitteasiantuntijat arvioivat kyseisten laitteiden jatkuvaksi maksimisuorituskyvyksi n. 5000 *Input/Output Operations Per Second* (IOPS) ja tämän hetkinen normaalkuorma laitteilla olisi n. 4000 IOPS arkipäivisin. Kuormitus siis jo normaalikäytöllä on melko ylärajoilla, eikä kovin paljoa pelivaraa suorituskyvyssä ole. Pidemmät isot kuormat ovat näkyneet huomattavina piikkeinä, jolloin suorituskyky on laskenut niin paljon, että asiaan on jouduttu puuttumaan.

Toimeksiantajaorganisaatiolle on kuitenkin jo hankittu uudet levylaittejärjestelmät; NetAppin valmistamat FAS3250 metroklusterit. Levyinä toimivat 2,5 tuuman, 600 GB:n, 10000 kierrosta / min, *Serial Attached SCSI* (SAS) –liitännällä olevat kiintolevyt. Pienempien levyjen etuja isompiin verrattuna ovat mm. isompi datatiheys, jolloin tiedonsiirtonopeudet kasvavat ja pienempi koko, joka mahdollistaa useamman kiintolevyn mahduttamisen rakkitoriiniin. Kyseiset järjestelmät pystyvät arviolta 12000 IOPS:n suorituskykyyn, joten itse levylaitteet eivät tule todennäköisesti olemaan pulonkaulana lähitulevaisuudessa.

9 Suorituskykymittaukset

9.1 Mittausväline

Suorituskykymittaukset suoritettiin organisaation työntekijän tekemällä pienellä Java-sovelluksella. Sen alkuperäinen tarkoitus oli vain testata onko palvelimen EJB-puoli hengissä. Myöhemmin siihen lisättiin vielä viiveen mittaus, jota tässäkin työssä käytettiin. Sovelluksen toiminta on hyvin yksinkertainen: kyselyn tullessa kasvatetaan EJB:ssä olevan laskurin arvoa yhdellä ja mitataan siihen mennyt aika. Mitään tietokantahakuja tai –kirjoituksia ei siis tällä sovelluksella voida tehdä, joten viiveet tulevat vain välimatkoista (esim. tietoverkko) ja prosessorin nopeudesta sekä sen mahdollisista muista kuormista. Kuviossa 8 on esitetty käytettävän testisovelluksen käyttöliittymä. Selaimen osoitekenttään laitettiin vain palvelimen osoite, joka toimii kyselijänä, ensimmäiseen kenttään kysyttävä palvelin ja portti sekä toiseen kenttään kyselykerrat. Tämän jälkeen painettiin Test-painiketta ja hetken päästä tyhjiin kenttiin tuli arvoja, joista tärkein oli viimeinen, joka kertoi kyselyihin menneen viiveen.



EJB 2.1 Tester

Enter direct EJB lookup location:

EJB method execution times:

Current bean location

Current bean value

Current bean object

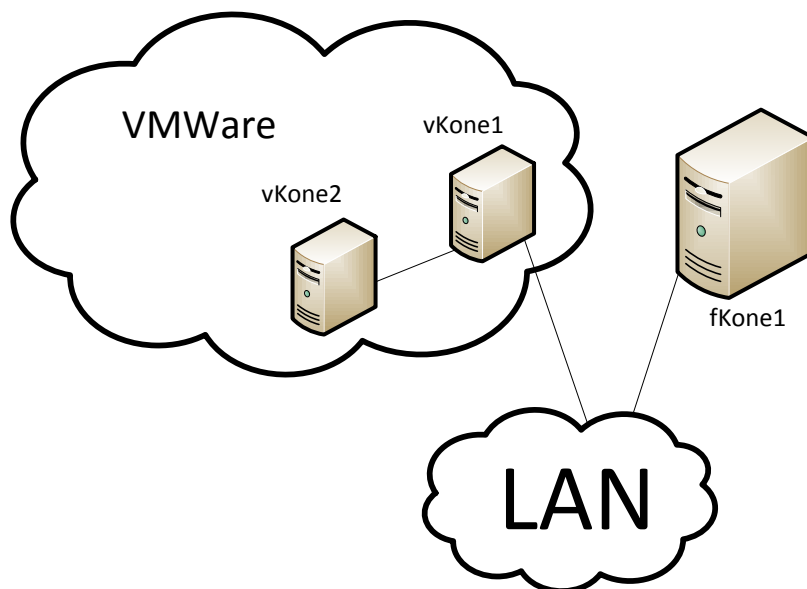
Current time lapse ms

Exceptions:

Kuvio 8. EJB-testeri

9.2 Testattavat palvelimet

Käytössä oli kolme testipalvelinta: kaksi virtuaalista (vKone1 ja vKone2) ja yksi fyysinen palvelin (fKone1). Virtuaalipalvelimet pyörivät molemmat samassa klusterissa ja fyysinen palvelin sijaitsi samassa konesalissa. Kuviossa 9 on havainnoistettu palvelimien suhteet toisiinsa. Testeissä mitattiin virtuaalikoneiden välisiä viiveitä ja miten ne erosivat, jos VM:t olivat samalla alustapalvelimella tai eri alustapalvelimilla. Tämän lisäksi kyselyitä tehtiin virtuaalikoneelta fyysiselle palvelimelle ja mitattiin viiveet. Viimeisenä testinä oli vertailu kumpi vastaa nopeammin itselleen: virtuaalinen (vKone2) vai fyysinen (fKone1) palvelin.



Kuvio 9. Testattavien palvelimien sijainnit toisiinsa nähden

Fyysisessä palvelimessa oli käytössä 24 GB keskusmuistia ja Intelin 4-ytiminen prosessori. Toisessa virtuaalipalvelimessa (vKone1) oli käytössä 4 GB keskusmuistia ja 1 vCPU ja toisessa (vKone2) 8 GB keskusmuistia ja 2 vCPU:ta.

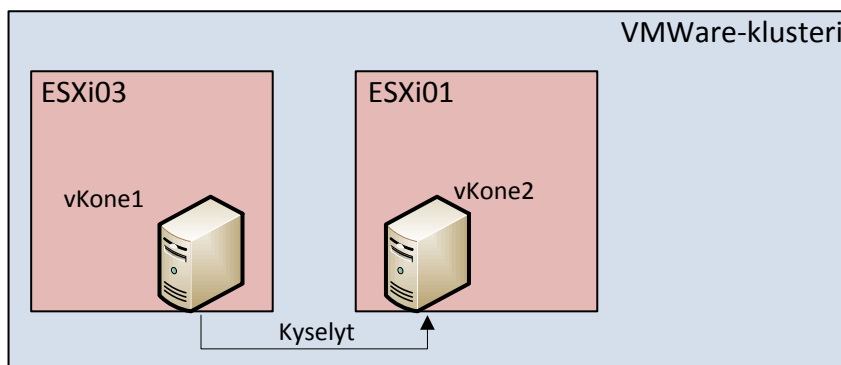
9.3 Testausmetodit

Koska toimeksiantajaorganisaatiolla ei vielä ollut erillisiä virtuaalialustoja testiympäristöön vaan testi- ja tuotantopalvelimet pyörivät samoilla alustoilla, kovin mittavia ja raskaita testejä ei voitu suorittaa. Ne ajettiin arkipäivänä klo 16 jälkeen, jotta palvelimet olisivat mahdollisimman vähällä käytöllä. Kyselykerroiksi asetettiin 10 000 ja kyselyt tehtiin 15 kertaa. Kysyvä palvelin oli aina vKone1, joka pysyi testien ajan samalla alustapalvelimella, paitsi kahdessa viimeisessä testissä, joissa vKone2 ja fKone1 –palvelimet tekivät kyselyt itselleen. Mittauspöytäkirja löytyy kokonaisuudessaan liitteestä 3.

9.4 Testit

9.4.1 Kysyvä ja kysyttävä palvelin eri alustapalvelimilla

Ensimmäisessä testissä kysyttävänä palvelimena toimi vKone2, joka sijaitsi ESXi01-alustapalvelimella ja kysyvä palvelin sijaitsi ESXi03:lla. Molemmat alustapalvelimet sijaitsivat samassa VMWare-klusterissa. Kuviossa 10 on esitetty tässä testissä käytetty topologia. Kyseiset alustapalvelimet olivat samassa klusterissa ja kyselyt menivät korttipalvelinkehikon nopean yhteyden läpi, eikä hitaampaa lähiverkkoa tarvinnut käyttää. Taulukossa 4 nähdään kyselyiden viiveet, keskiarvo ja keskihajonta.



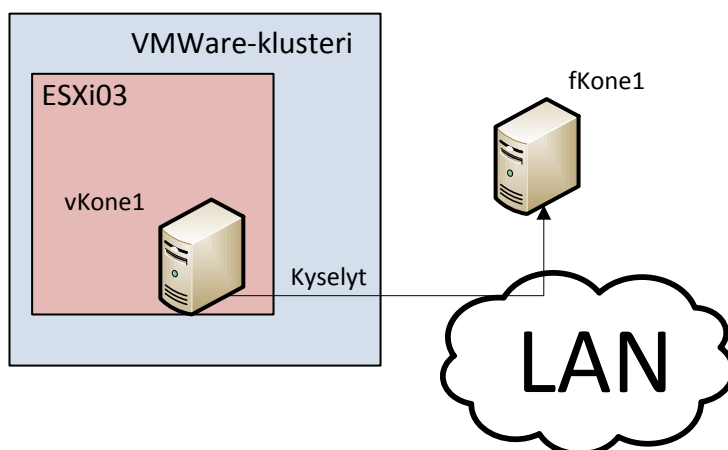
Kuvio 10. Ensimmäisen testin topologia

Taulukko 4. Kyselyviiveet eri alustapalvelimien välillä

Kysely nro.	Aika (ms)
1	6040
2	6618
3	8158
4	6207
5	6671
6	8126
7	8309
8	6651
9	9343
10	8826
11	8440
12	6935
13	7517
14	6281
15	8260
Keskiarvo	7492
Keskihajonta	1065

9.4.2 Kyselyt virtuaaliselta palvelimelta fyysiselle

Tässä testissä data liikkui VMWaren alustapalvelimista ulkopuolelle, joten viivettä tuli verkon puolelta lisää. Kuviossa 11 on esitetty testissä käytettävä topologia. Viiveiden keskiarvo ei kuitenkaan huomattavasti noussut verrattuna edelliseen testiin, kuten taulukossa 5 huomataan, mutta keskihajonta nousi 40 %, mikä saattaa kertoa lähi-verkon tai VMWaren ulkoisten verkkoliitäntöjen kuormituksesta.



Kuvio 11. Toisen testin topologia

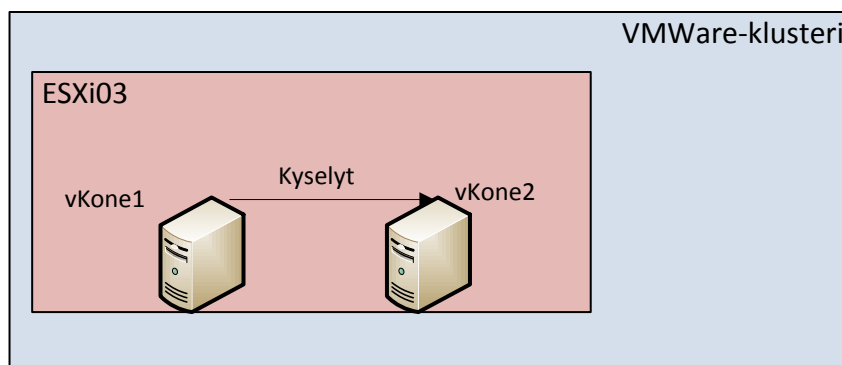
Taulukko 5. Kyselyviiveet virtuaaliselta palvelimelta fyysiselle

Kysely nro.	Aika (ms)
1	5633
2	8261
3	8319
4	11293
5	10522
6	8410
7	7321
8	6598
9	7196
10	7298
11	7027
12	7046
13	7892
14	6484
15	6802
Keskiarvo	7740
Keskihajonta	1495

9.4.3 Kyselyt saman alustapalvelimen sisällä VM:ltä toiselle

Tähän testiin kysyvä ja kysytty palvelin siirrettiin samalle fyysiselle VMWare-alustapalvelimelle, joka oli ESXi03. Kuvio 12 havainnollistaa testissä käytettävän topologian. Keskihajonta verrattuna eri alustapalvelimien välisille kyselyille ei juurikaan

muttunut, mutta keskiarvo tippui lähes 12 %. Taulukossa 6 käy ilmi kyseisen testin tulokset. Tästä voidaan päätellä, että toisiinsa yhteydessä olevat virtuaalipalvelimet kannattanee laittaa samalle alustapalvelimelle. Ainakin tässä kyseisessä testissä siitä oli merkittävä hyöty.



Kuvio 12. Kolmannen testin topologia

Taulukko 6. Kyselyviiveet saman alustapalvelimen sisällä

Kysely nro.	Aika (ms)
1	6025
2	6392
3	8410
4	5585
5	6098
6	6800
7	5583
8	5472
9	6100
10	6459
11	5647
12	8061
13	7911
14	7675
15	7087
Keskiarvo	6620
Keskihajonta	989

9.4.4 Kyselyt VM:n sisällä

Toiseksi viimeisessä testissä mitattiin viiveitä virtuaalikoneen sisällä eli kysyvä ja kysyttävä palvelin olivat sama. Taulukossa 7 nähdään mittaustulokset ja yhdellä mittauuskerralla viive oli n. kolminkertainen verrattuna muiden tulosten keskiarvoon, joten se jätettiin huomioimatta.

Taulukko 7. Kyselyviiveet virtuaalikoneen sisällä

Kysely nro.	Aika (ms)
1	332
2	340
3	344
4	347
5	337
6	345
7	342
8	368
9	348
10	350
11	336
12	1080
13	341
14	337
15	334
Keskiarvo	343
Keskihajonta	9

9.4.5 Kyselyt fyysisen palvelimen sisällä

Taulukossa 8 nähdään fyysisen palvelimen sisällä tehtyjen kyselyiden viiveet. Siitä huomataan, että keskiarvo on lähes 40 % pienempi kuin virtuaalisen koneen sisällä tehdyt kyselyt. Vaikka molemmilla palvelimilla on suurin piirtein yhtä tehokkaat prosessorit alla, ovat tulokset huomattavasti fyysisen palvelimen puolelle kallistuneita. Virtuaalikoneiden vCPU:t kärsivät suorituskyyvyssä verrattuna suoraan käytössä oleviin fyysisiin prosessoreihin ainakin tässä testissä.

Taulukko 8. Kyselyviiveet fyysisen palvelimen sisällä

Kysely nro.	Aika (ms)
1	247
2	197
3	197
4	190
5	194
6	216
7	194
8	195
9	194
10	224
11	199
12	199
13	200
14	195
15	300
Keskiarvo	209
Keskihajonta	29

9.5 Tuloksien pohdinta

Saadut mittaustulokset olivat hyvin pitkälle odotettavissa. Suurimmat yllätykset olivat ehkä virtuaalikoneen ja fyysisen palvelimen välisien kyselyiden viiveet verrattuna virtuaalikoneiden välisiin kyselyihin, mutta ne selittynee verkkoviiveillä ja sisäisten kyselyiden viive-erot virtuaalikoneen ja fyysisen koneen sisällä. En itse osannut odottaa 40 %:n eroa fyysisen koneen hyväksi, vaikka pientä etua fyysiselle palvelimelle odotin.

Yksi mielenkiintoisista asioista oli, kun siirrettiin kysyvä ja kysytty VM samalle alustapalvelimelle, tippui vastausten keskiviive n. 12 %. Tämä on mielestäni yksi asia, johon kannattaa kiinnittää huomiota suunnitellessa ja käyttöönottaessa tai jopa optimoimassa jo olemassa olevia isoja virtuaalijärjestelmiä, joissa useat palvelimet ovat yhteydessä toisiinsa.

Jos testeistä olisi haluttu ”täydelliset”, olisi mukana pitänyt olla tietokantahakuja.

Tähän ei kuitenkaan pystytty organisaation tietojärjestelmien rakenteen vuoksi, jossa

ei ole erillistä testiympäristöä virtuaali- ja levyjärjestelmäympäristössä. Fyysiset testipalvelimet ovat erikseen, mikä onkin helppo toteuttaa verrattuna edellä mainittuihin. Testikäytössä olevat virtuaalipalvelimet sijaitsevat samoilla levyjärjestelmillä ja VMWaren alustapalvelimilla kuin tuotannossa olevat virtuaalipalvelimetkin. Tämän lisäksi vanha levyjärjestelmä on lähes maksimikuormalla koko ajan. Näiden syiden takia mittavia I/O-testejä ei voitu suorittaa ja jouduttiin tyytymään hieman kevyempiin menetelmiin. Näillä testeillä ei käyty tietokannoissa (levyjärjestelmissä) tekemässä hakuja kertaakaan vaan kyselyt olivat prosessoreiden ja muistien välisiä laskureiden arvojen kasvattamisia.

10 Yhteenveto

10.1 Työn tulokset ja niiden arviointi

Työn tavoitteena oli vertailla virtuaalisten ja fyysisten palvelimien eroa sovellusalustakäytössä sekä luoda näistä toimeksiantajalle kirjallinen dokumentti, sillä heillä ei aikaisemmin ole mitään kirjallista vertailua ollut, vaan lähes kaikki tieto on ollut ”asiantuntijoiden päässä”. Dokumenttina toimii tässä tapauksessa valmis opinnäytetyö, jossa on käyty kyseisten palvelimien eroja teoriassa ja käytännön suorituskäytösteissä. Työssä käytiin läpi myös kustannuseroja, vikatilanteissa toimimista ja minkälaisia pullonkauloja virtuaalisesta ympäristöstä löytyy.

Pullonkauloista mielenkiintoisimpana oli x86-arkkitehtuurin rajoitteet I/O-painotteisissa tehtävissä. Tästä ei juurikaan löytynyt internetistä eikä kirjallisuudesta mainintaa virtualisoinnin kohdalla. Käytännön kokemuksia kuitenkin toimeksiantajan puolelta löytyi ja lähdinkin sen pohjalta asiaa tutkimaan. Kävi ilmi, että välimuistipohjainen ratkaisu voi joissain tapauksissa olla jarruttava tekijä, eikä ratkaisuna toimi pelkkä raa’an laskuvoiman lisäys tai levyjärjestelmän vaihto.

Työn alussa oli vielä epäselvää kuinka käytännön testit suoritetaan, mutta lopulta ne päätettiin suorittaa toimeksiantajan testipalvelimilla. Harmittavaa kuitenkin oli, että sen hetken ympäristössä virtuaaliset testipalvelimet sijaitsivat samoilla alustapalvelimilla kuin tuotantopalvelimetkin, joten mitään raskaita tai laajoja testejä ei voitu suorittaa. Suorituskäytösteissä mielenkiintoisimmat havainnot liittyivät virtuaalipalvelimien välisiin kyselyihin. Ainakin näissä testeissä oli huomattava ero sillä, että olivatko virtuaalipalvelimet samalla alustapalvelimella vai eri alustoilla saman klusterin sisällä. Kaikki saman klusterin alustapalvelimet olivat samassa korttipalvelinkehikossa.

Mielestäni työ onnistui melko hyvin loppupään sähläyksistä huolimatta, kun käyttäjätunnukseni toimeksiantajaorganisaation puolelta menivät lukkoon työ sopimuksen päättymisen takia, eikä niitä enää saatu auki. Käytännön testit oli onneksi kuitenkin

tehty ennen sitä ja otettu talteen ulkoiselle medialle, joten työ saatiin tehtyä loppuun.

10.2 Kehittämiskohteet

Seuraava kehityskohde levyjärjestelmien uusimisen jälkeen mielestäni olisi virtuaaliympäristön uudelleenrakentaminen. Nykyinen järjestelmä on rakennettu lähinnä testikäyttöön ja siitä uusia alustapalvelimia lisäämällä laajennettu myös tuotantokäyttöön. Sitä suunniteltaessa ei oltu osattu ottaa huomioon suurta I/O-kuormaa, jota organisaation tietojenkäsittely suurimmaksi osaksi on. Tällä hetkellä ongelma on ratkaistu siirtämällä suurimmat kuormat aiheuttavat palvelimet virtuaalialustoilta fyysisille palvelimille. Hyvin suunniteltu ja toteutettu virtuaaliympäristö on huomattavasti helpompi ylläpitää, kustannustehokkaampi ja skaalautuvampi, joten siihen pitäisi mielestäni pyrkiä.

Yksi vaihtoehto olisi vaihtaa nykyiset järeät, 2 fyysisen prosessorin sisältävät, alustapalvelimet useampaan yhden fyysisen prosessorin vastaavaan. Tällä tavalla virtuaalipalvelimia voidaan laittaa vähemmän alustapalvelinta kohden ja täten kuorma jokaisella vähenee. Tämä voisi auttaa nykyiseen I/O-ongelmaan huomattavasti. Sen lisäksi voitaisiin miettiä olisiko hyötyä laittaa toisiinsa paljon yhteydessä olevat virtuaalipalvelimet pyörimään samalle alustapalvelimelle.

Organisaatio voisi miettiä myös enemmän vapaan lähdekoodin ohjelmistojen käyttämistä. Kunhan uudet virtuaaliympäristöt joskus hankitaan ja testipuoli saadaan eri alustoille kuin tuotantopuoli, voitaisiin tehdä käytännön testejä kuinka jokin vapaan lähdekoodin sovellus toimisi heidän käytössä kaupallisen sovelluksen sijaan. Open source –tuotteilla kuitenkin saataisiin suuret säästöt hankintahinnoissa ja toiminnallisuudeltaan ne ovat nykyään yleensä samalla viivalla kaupallisten sovelluksien kanssa.

Lähteet

Bayo, J.G., Fuchs, I., Korn, G., Rebolj, R., Seidel, V. & Tretau, R. 2003. The IBM Total-Storage NAS Integration Guide. USA: IBM Redbooks.

Beck, P., Ibarra, H.H., Kuvaravel, S., Miklas, L. & Tate, J. 2012. Introduction to Storage Area Networks and System Networking. USA: IBM Redbooks.

Campbell, S. & Jeronimo, M. 2006. Applied Virtual Technology - Usage models for IT professionals and Software Developers. USA: Intel Press.

Evans, I. 2013. Your First Cup: An Introduction to the Java EE Platform, Volume 1, Release 7 for Java Platform, Enterprise Edition.

<http://docs.oracle.com/javaee/7/firstcup/doc/firstcup.pdf>

Fleming, M. & Perry, R. 2010. Business Value of JBoss Enterprise Application Platform. Viitattu 25.11.2013.

http://www.redhat.com/f/pdf/IDC_ROI_study_JBoss_Enterprise_Application_Platform.pdf

Herold, S., Laverick, M. & Oglesby, R. 2008. VMWare Infrastructure 3: Advanced Technical Design Guide and Advanced Operations Guide. Mattoon, IL: United Graphics.

Java EE 5 Tutorial. 2007. Oraclen laatima tutoriaali Java EE 5 –palvelinympäristöön.

Viitattu 21.12.2013. <http://docs.oracle.com/javaee/5/tutorial/doc/docinfo.html>

JBoss vs. WebSphere – A TCO Analysis. 2011. Virtuant Corpin laatima vertailu. Viitattu 25.11.2013. <http://www.redhat.com/jboss/getunstuck/Virtuant-TCO-Analysis-JBossEAP-vs-Websphere.pdf>

Lowe, S. 2011. Masterin VMware vSphere 5. Indianapolis, Indiana: John Wiley & Sons Inc.

PVU. n.d. Processor Value Unit [PVU] licensing for Distributed Software. Päivitetty 28.10.2013. Viitattu 26.11.2013. [http://www-](http://www-01.ibm.com/software/lotus/passportadvantage/pvu_licensing_for_customers.html)

[01.ibm.com/software/lotus/passportadvantage/pvu_licensing_for_customers.html](http://www-01.ibm.com/software/lotus/passportadvantage/pvu_licensing_for_customers.html)

Stallings, W. 2009. Operating Systems: Internals and Design Principles, 6th edition. Upper Saddle River, NJ: Pearson Education, Inc.

Understanding full virtualization, paravirtualization, and hardware assist. 2007. Viitattu 7.10.2013. <http://vmware.com>

Virtualizing Server Workloads. 2008. AMD White Paper. Viitattu 15.10.2013.

<http://www.amd.com>

VMware vSphere Basics. n.d. VMware vSpheren tuotedokumentaatio. Viitattu 23.10.2013. <http://pubs.vmware.com>

WAS system requirements. 2013. IBM:n antamat laitevaatimukset WAS:lle. Viitattu 11.12.2013. <http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg27006921>

WAS V8.5 vs. Jboss EAP v6 TCO Analysis. 2012. Profilicsin laatima vertailu. Viitattu 2.12.2013. ftp://public.dhe.ibm.com/software/webervers/appserv/pdf/wp_WAS_JBoss_TCO_Analysis.pdf

Liitteet

Liite 1. Virtuant Corpin tutkimuksen kustannuserot WAS ND 7:n ja JBoss EAP 5:n välillä

Kategoria	Osio	WAS ND 1. vuosi	3 vuoden TCO	Jboss EAP 1. vuosi	3 vuoden TCO
Laitteisto					
	Palvelimien lkm.	9		9	
	Hinta	\$15,900		\$15,900	
	YHT.	\$143,100	\$143,100	\$143,100	\$143,100
Lisenssit					
	Palvelimien lkm.	9		9	
	Lisenssin hinta / palvelin	\$103,040		\$0	
	YHT.	\$927,360	\$927,360	\$0	\$0
Ohjelmistotuki		% lisenssistä			
	Ylläpito	17.5%		\$0	
	Vuositilaus	5.0%		\$51,250	
	YHT.	\$208,656	\$625,968	\$51,250	\$153,750
Monitorointi ja hallinta (Tivoli & JON)					
	Hankinta	\$696,960		\$0	
	Vuositilaus	\$34,848		\$0	
	YHT.	\$731,808	\$801,504	\$0	\$0
Ohjelmistoasiantuntijat					
	Asiantuntijoiden lkm.	5		5	
	Vuosipalkka	\$102,000		\$92,000	
	Käyttöönottoasiantuntija	1		1	
	Vuosipalkka	\$65,000		\$65,000	
	YHT.	\$575,000	\$1,725,000	\$525,000	\$1,575,000
Infrastruktuurin hallinta					
	Pääkäyttäjien lkm.	2		2	
	Vuosipalkka	\$95,000		\$78,500	
	YHT.	\$190,000	\$570,000	\$157,000	\$471,000
Koulutus					
	Henkilöiden lkm	8		8	
	Koulutuksia vuodessa	2		1	
	Hinta per henkilö	\$3,400		\$2,800	
	YHT.	\$54,400	\$163,200	\$22,400	\$67,200

Sovelluskehitys

Tuntipalkka	\$51		\$46	
Projekteja vuodessa	3		3	
Projektin tuntimäärä	450		300	
YHT.	\$68,850	\$205,550	\$41,400	\$124,200
LOPPUSUMMA	\$2,899,174	\$5,162,682	\$940,150	\$2,534,250
Ero dollareina			-\$1,959,024	-\$2,628,432
Ero prosentteina			-67.57 %	-50.91 %

Liite 2. Prolificsin tutkimuksen kustannuserot WAS V8.5:n ja Jboss EAP v6:n välillä

Kategoria	WAS ND 5 vuoden TCO	Jboss EAP 5 vuoden TCO
Laitteisto	\$2,060,934	\$3,114,308
Koulutus	\$84,375	\$171,998
Ohjelmistolisenssit	\$2,623,920	-
Ohjelmistotuki	\$2,008,815	\$1,821,316
Sovelluksen hallinta	\$759,492	\$2,570,500
Infrastruktuurin hallinta	\$1,533,834	\$2,301,566
Riskit ja häiriöaika	-	\$2,268,548
YHT.	\$9,071,370	\$12,248,235
Ero dollareina	-\$3,176,865	
Ero prosentteina	-25.94 %	

Liite 3. Suorituskykymittauspöytäkirja

Kysyvä	Kysytty	Aika (ms)															Keskia- vo (ms)	Keskiha- jonta (ms)
vkone1 (ESXi03)	vkone2 (ESXi01)	6040	6618	8158	6207	6671	8126	8309	6651	9343	8826	8440	6935	7517	6281	8260	7492	1065
vkone 1 (ESXi03)	fkone1	5633	8261	8319	11293	10522	8410	7321	6598	7196	7298	7027	7046	7892	6484	6802	7253	1495
vkone1 (ESXi03)	vkone2 (ESXi03)	6025	6392	8410	5585	6098	6800	5583	5472	6100	6459	5647	8061	7911	7675	7087	6620	989
vkone2 (ESXi03)	vkone2	332	340	344	347	337	345	342	368	348	350	336	4080	341	337	334	343	9
fkone1	fkone1	247	197	197	190	194	216	194	195	194	224	199	199	200	195	300	209	29